1. Definition

The Dining Philosophers Problem is a computer science problem that relates to synchronization and resource allocation in concurrent programming. This problem illustrates a scenario in which five philosophers sit around a dining table. There are five plates of spaghetti on the table, each in front of a philosopher, and each philosopher needs two forks to eat.

Each philosopher needs both their left-hand fork and right-hand fork to eat, but there are only five forks available, with one between each pair of philosophers. The philosophers can either think or eat, but they can't do both at the same time. When a philosopher finishes eating, they will put down both forks. Thus, a philosopher can only eat when their two neighbors are thinking instead of using any fork adjacent to him. No philosopher should starve – in other words, endlessly thinking.

The challenge here is to design an algorithm or a strategy to ensure that the philosophers can eat without creating a deadlock, where everyone is waiting for a fork that will never be available. Solutions to this problem need to consider preventing deadlocks and offering every philosopher a chance to eat without creating conflicts with neighboring philosophers.

1. Solution

To solve this problem, first we need to make sure that no two philosophers can pick up the same fork at the same time. Semaphore is considered a prerequisite to achieve this in concurrent programming. It essentially refers to an integer variable to control whether a thread is eligible to access resources. In our case, a binary semaphore is used to create a critical section, a protected section that cannot be accessed by more than one process or thread (“Critical section”, 2023). We also need an array of semaphores to represent each fork, with all the elements initialized as 1 to indicate availability. Lastly, there should be an array of state variables (thinking/hungry/eating) to represent each philosopher. To make sure the program works, a philosopher is only allowed to pick up their nearest forks and eat when both forks are available, otherwise this might lead to a starvation as each philosopher holds their left fork thinking endlessly, refusing to put down.

For each philosopher, assign them a process in which they conduct the following: First, they think for a certain amount of time, and then they get hungry (change their status from thinking to hungry) and would try to pick up forks adjacent to them. We let them enter the critical section where they test whether their neighboring philosophers are eating. If not, they’ll change their status to eating for an amount of time while locking the corresponding forks using semaphores, else they’ll just wait to be signaled. When they finish eating, they release the semaphores representing those forks and remind their neighbors to conduct the same behaviors before restoring their state back to thinking. This strategy ensures that every philosopher has the opportunity to think and eat alternatively while no one would starve forever.

1. Implications

In this problem, philosophers represent processes and forks represent limited resources that must be shared between processes in a synchronized manner without violating any rules. This problem effectively illustrates the concept of shared resources and deadlock in concurrent programming, and solutions to it utilize a variety of synchronization techniques including mutex and semaphores. It also sheds light on large-scale computer applications such as database management systems where multiple transactions attempt to access and modify shared data concurrently. Overall, it is a great learning experience for those who want to harness concurrent programming.

Reference:

Critical section. (n.d.). In *Wikipedia*. Retrieved October 9, 2023, from <https://en.wikipedia.org/wiki/Critical_section>

Dining philosophers problem. (n.d.). In *Wikipedia*. Retrieved October 9, 2023, from <https://en.wikipedia.org/wiki/Dining_philosophers_problem>

Dining Philosopher Problem Using Semaphores. (n.d.). In *Geeks for Geeks*. Retrieved October 9, 2023, from <https://www.geeksforgeeks.org/dining-philosopher-problem-using-semaphores>